ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 4026 – 4030

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

A New Evolutionary Computation Based Approach for Learning Bayesian Network

Yungang Zhu[a,b], Dayou Liu[a,b]*, Haiyang Jia[a,b]

[a]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[b]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China*

Abstract

Bayesian network is a popular tool for uncertainty process in Artificial Intelligence. In recent years, more and more attention has been paid to learning of Bayesian network. In this paper, we proposed a novel learning algorithm for Bayesian network based on (μ, λ)-Evolution Strategy, we present the encoding scheme and fitness function, designed the evolutionary operators of recombination, mutation and selection. Theoretical analysis and experimental results all demonstrate that the proposed method can learn the Bayesian network from data effectively.

Keywords: Bayesian network; Evolution strategy; Evolutionary computation

1. Introduction

Bayesian network has been a powerful tool for managing uncertainty. It has been successfully applied to expert system, diagnosis system, and decision support system et al. Bayesian network integrates graphical model and probability theory, and it indicates the internal relationship among variables.

In recent years, many researchers pay much attention to the learning algorithm for Bayesian network. Learning the structure of a Bayesian network can be considered a specific example of the general problem of selecting a probabilistic model that explains a given set of data[1].

* Corresponding author.
E-mail address: dyliu@jlu.edu.cn.

doi:10.1016/j.proeng.2011.08.755

In this paper we proposed a novel learning algorithm for Bayesian network based on Evolution Strategy. In detail, we adopt "(μ, λ)-Evolution Strategy" which is an evolutionary computing method to learn Bayesian network.

The rest of this paper is organized as follows: Section 2 presents brief background knowledge. The method we proposed is described in Section 3. Section 4 presents the experimental results and discussion of the proposed method. Finally, conclusions are summarized in Section 5.

2. Theoretical Background

A Bayesian network (BN) is a graphical model for representing relationships among variables. Let us consider a set of variables $X=\{X_1,X_2,\ldots,X_n\}$, a Bayesian network is a tuple (G, Θ), where G is a directed acyclic graph (DAG), each node of G represents the variable, and each directed edge represents relationships between variables; and $\Theta=\{ P(X_i \mid \pi_i), 1\le i\le n \}$ represents the local conditional probability distribution of each node given the values of their parent nodes, where π_i is the parent set of X_i.

Assumed the range of X_i is $\{ x_i^1,\ldots,x_i^{r_i} \}$, the range of π_i is $\{ \pi_i^1,\ldots,\pi_i^{q_i} \}$, the local conditional probability distribution of each node is represented by $\theta_{ijk} = P(X_i = x_i^k \mid \pi_i = \pi_i^j)$, and $\Theta = \bigcup_{i=1}^{n}\bigcup_{j=1}^{q_i}\bigcup_{k=1}^{r_i}\{\theta_{ijk}\}$.

Evolution Strategy (ES) is one of the evolutionary computing methods. ES produces consecutive generations of individual, during a generation a selection method is used to select specific individuals which form the new generation by recombination and mutation [2].

3. Learning Bayesian network based on Evolution Strategy

The problem of learning of Bayesian network can be stated as follows. Assuming that D represent the data, the purpose is to obtain a Bayesian network S that best fit the D.

3.1. Encoding

The Bayesian network structure was encoded into adjacency matrix or adjacency list in previous coding way, but the method will result in a large amount of cyclic graphs which are illegal structures. Based on the coding scheme in [3][4], in this paper, the code is divided into 3 parts.

The 1st part is a sequence of nodes: This order is the reverse of topological sort of the network nodes, so there is no cycle. For example, the sequence of Figure 1 is 54231.

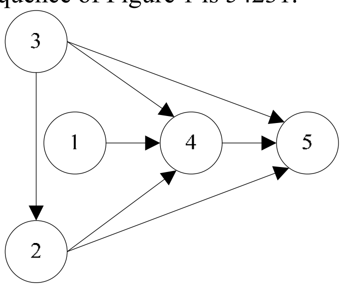

Fig. 1. A simple example of Bayesian network.

The 2nd part has n-1 segments, each segment indicate the parents of each node in the sequence above, the last node has no parent, so only n-1 segments needed. For example, the structure of Figure 1, the second part of code is 1110 111 10 0. segment 1 indicate that for node 5, node 4,2,3 is parent, and Node 1 is not parent; segment 2 indicate that for node 4, node 2,3,1 is parent; segment 3 indicate for Node 2, Node 3 is parent, and Node 1 is not; and so on. Further more, considering simple of network structure, we restrict the number of parents of each node can not exceed k. In general, k is much smaller than n, and using the code above will include lots of 0. So we use compress form of the code, that is only recording the position of 1 in the order of sequence from first part, thus the final code is [234 345 4 0].

The 3rd Part is adaptive step size in mutation evolutionary strategy σ. So in summary, the code of Figure 1 is [54231 234 345 4 0 σ].

3.2. Fitness function

We use Bayesian Information Criterion (BIC)[1] scoring measure to be the Fitness function , that is as follows:

$$Fitness(S) = \log P(D \mid S) - Pen(S) = \sum_i \left(\sum_j \sum_k (N_{ijk} \log \theta_{ijk}) - \frac{1}{2} \log L \cdot |\pi_i| (|X_i| - 1) \right)$$

where N_{ijk} is number of sample which $X_i = X_i^k$ and $\pi_i = \pi_i^j$ in data D. L is the number of samples. $|X|$ is the number of assignment of *X*. and $P(D \mid S)$ measures the fitness of S to data *D*, Pen(*S*) is penalty function about the structure of *S* to make the learning algorithm trend to obtain concise model which is easy for management.

3.3. Evolutionary operator

- **Recombination**

The Recombination of evolution strategy is equivalent of the cross for genetic algorithm. But unlike GA, the Recombination generates only one individual from two parent individuals.

For the first part of the code, we use Partially Matched Crossover of GA[5], selecting a new individual from the two resulting individuals randomly. For the second part of the code, for each segment, a segment is selected randomly from the two parent individuals as the segment of the child individual. The third part is the size of step in Mutation, using the mid-value for Recombination. If the third part of two parents are $\sigma_1$and σ_2, after Recombination, it is $(\sigma_1+\sigma_1)/2$.

- **Mutation**

There are three types of mutation operator: the adding an arc, deleting an arc, and reversing an arc. It is not to perform one mutation operator during mutation, but to perform $|\sigma \cdot N(0,1)|$ mutation operators, $N(0,1)$ is normally distributed random variable which mean=0 and variance=1.

- **Selection**

Selection is strictly according to the fitness, eliminating all the poor individuals, selecting all the good ones. The proposed algorithm using (μ, λ) selection strategy: μ parent individuals generate λ (λ>μ) children individuals, and select μ individuals from the resulting λ individuals as the next generation.

In summary, the pseudo-code of the learning algorithm based on $(\mu, \lambda) - ES$ described as follows:

Procedure *ESBN* (data *D*)
begin
  Generate μ Bayesian networks randomly as init group S(0);
  Select a network randomly from S(0) as current best network S_{max};
  for each S_i **in** $S(0)$ **do**
    Fitness[S_i]= *Cal-Fitness*(S_i) ; //Calculate Fitness
  end for

S_{max}=*Select_Top_One*(S(0)) ;
S_{min}=*Select_Lowest_One*(S(0)) ;
t=0;
while ($t< t_{max}$ || $Fitness[S_{max}] - Fitness[S_{min}] < \varepsilon$) **do**
for i=1 **to** λ **do** //Generate λ children individuals
$S_i(t) \leftarrow$ *Random_Select*($S(t)$) ;
$S_j(t) \leftarrow$ *Random_Select*($S(t)$) ;
$Children_i \leftarrow$**Recombination**($S_i(t)$, $S_j(t)$) ; // Recombination
$Children_i \leftarrow$**Mutation**($Children_i$) ; // Mutation
Fitness[$Children_i$]=*Cal-Fitness*($Children_i$) ;
end for
Select μ individuals as next generation $S(t+1)$ from // Selection
{$Children_1$,…, $Children_\lambda$} according to Fitness[$Children_i$] ($1 \leqslant i \leqslant \lambda$)
$t=t+1$;
S_{max}=*Select_Top_One*($S(t)$);
S_{min}=*Select_Lowest_One*($S(t)$);
end while
return S_{max} ;
end █

4. Experiment and discussion

We use benchmark experiment data generated from a classical Bayesian network called Alarm [6] which has 37 nodes. In detail, we generate a training data set with 4000 samples, the first 3000 samples are used for learning, and the last 1000 samples are used for testing. We make the learning samples into 3 groups each of which contains 1000 samples, 2000 samples, 3000 samples, then we learn 3 Bayesian networks from the 3 groups and test the learning accuracy separately. The algorithm is evaluated based on the average Log-Loss of each learned network on this test set, that is $\frac{1}{N}\sum_{i=1}^{N} P_S(C_i)$ [7], where N is the number of test data, C_i is the ith sample of test data. For convenience, we actually use the absolute value of Log-Loss, it's value can measure how well the learned network fit the data, that is the accuracy of learned network, and the value is the smaller, the better. The results are summarized in Figure 2.

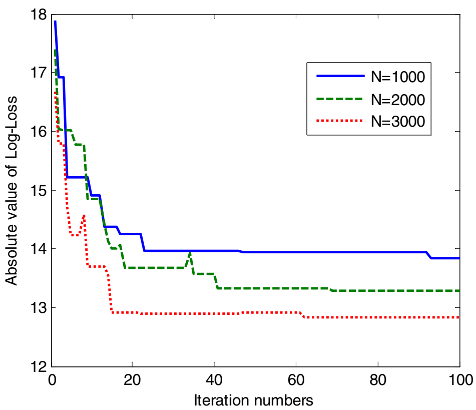

Fig. 2. The learning performance of the proposed algorithm with 3 groups test data.

From Figure 2, we can see the algorithm can converge to a good network, and the more samples used for learning, the faster algorithm will converge, and the better Bayesian network will be obtained. Because more data can contains more statistical features, so the learned result will be more accurate. The experimental shows that the algorithm is effective.

5. Conclusions

In this paper, a (μ, λ)-Evolution Strategy based learning algorithm for Bayesian network is proposed. An improved encoding scheme of Bayesian network structure is proposed, the fitness function is designed based on the BIC scoring measure. The recombination, mutation and selection evolutionary operators are also proposed. Experimental results show that the proposed algorithm can learn the Bayesian network from data effectively.

Acknowledgements

Professor Dayou Liu is the corresponding author for this paper. This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No.60873149, 60973088, 60773099. This work is also supported by the Open Projects of Shanghai Key Laboratory of Intelligent Information Processing in Fudan University under the Grand No. IIPL-09-007.

References

[1] Daly R, Shen Q, Aitken S. Learning Bayesian networks: approaches and issues.*The Knowledge Engineering Review*,2011, 26(2), p99–157.

[2] Gruttner M, Sehnke F, et al. Multi-Dimensional Deep Memory Atari-Go Players for Parameter Exploring Policy Gradients. *The 20th International Conference on Artificial Neural Networks* ,2010, p114-123.

[3] Zhang C, Shen YD, et al. Structure Learning of Belief Network by Genetic Algorithms: A New Network Encoding Method. *Computer Science*.2004, 31(12),p103-105.

[4] Lee J, Chung W,et al.A new genetic approach to structure learning of Bayesian networks.*Advances in Neural Networks - ISNN 2006, PT 1 Lecture Notes in Computer Science* 3971, Part 1 2006, p659-668.

[5] Zhou CJ, Liang YC. *Evolutionary Computation*. 3rd ed. Changchun:Jilin University Press;2009.

[6] Beinlich I, Suermondt G, et al. The ALRAM monitoring system: A case study with two probabilistic inference. *The 2nd European Conf on A rtificial Intelligence in Medicine*, 1989, p247-256.

[7] Friedman N. Learning belief networks in the presence of missing values and hidden variables.*The 14th International Conf on Machine Learning* ,1997, p125-133.